ORIGEN: Zero-Shot 3D Orientation Grounding in Text-to-Image Generation

Yunhong Min* **Daehyeon Choi*** **Kyeongmin Yeo** **Jihyun Lee** **Minhyuk Sung**
KAIST
{dbsghd363,daehyeonchoi,aaaaa,jyun.lee,mhsung}@kaist.ac.kr

Figure 1: **3D orientation-grounded text-to-image generation results of ORIGEN.** We present ORIGEN, the first zero-shot method for 3D orientation grounding in text-to-image generation across multiple objects and diverse categories. ORIGEN generates high-quality images that are accurately aligned with the grounding orientation conditions (colored arrows) and the input text prompts.

Abstract

We introduce ORIGEN, the first zero-shot method for 3D orientation grounding in text-to-image generation across multiple objects and diverse categories. While previous work on spatial grounding in image generation has mainly focused on 2D positioning, it lacks control over 3D orientation. To address this, we propose a reward-guided sampling approach using a pretrained discriminative model for 3D orientation estimation and a one-step text-to-image generative flow model. While gradient-ascent-based optimization is a natural choice for reward-based guidance, it struggles to maintain image realism. Instead, we adopt a sampling-based approach using Langevin dynamics, which extends gradient ascent by simply injecting random noise—requiring just a single additional line of code. Additionally, we introduce adaptive time rescaling based on the reward function to accelerate convergence. Our experiments show that ORIGEN outperforms both training-based and test-time guidance methods across quantitative metrics and user studies. Project Page: https://origen2025.github.io.

1 Introduction

Controllability is a key aspect of generative models, enabling precise, user-driven outputs. In image generation, *spatial grounding* ensures structured and semantically meaningful results by incorporating conditions that cannot be fully specified through text alone. Recent research integrating spatial instructions, such as bounding boxes [1–5] and segmentation masks [3, 6, 7, 4, 8, 9], has

*Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

shown promising results. While these works have advanced 2D spatial control, particularly *positional* constraints, 3D spatial grounding remains largely unexplored. In particular, orientation is essential for defining an object's spatial pose [10–16], yet its integration into conditioning remains an open challenge.

A few existing methods, such as Zero-1-to-3 [17] and Continuous 3D Words [18], support orientation-conditioned image generation. However, Zero-1-to-3 enables only relative orientation control with respect to a reference foreground image, while Continuous 3D Words is limited to single-object images and supports only half-front azimuth control. Moreover, all these models lack realism because they are trained on synthetic data, i.e., multi-view renderings of centered 3D objects, as real-world training images with accurate per-object orientation annotations are not publicly available. In addition, OrientDream [19] supports orientation control via text prompts, but it is restricted to four primitive azimuths (front, left, back, right) and is also limited to single-object images.

To overcome these limitations, we propose ORIGEN, the first method for generalizable 3D orientation grounding in real-world images across multiple objects and diverse categories. We introduce a *zero-shot* approach that leverages test-time guidance from OrientAnything [20], a foundational discriminative model for 3D orientation estimation. Specifically, using a pretrained one-step text-to-image generative model [21] that maps a latent vector to a real image, along with a reward function defined by the discriminative model, our goal is to find a latent vector whose corresponding real image yields a high reward.

A natural approach for this search is gradient-ascent-based optimization [22], but it struggles to keep the latent distribution aligned with the prior (a standard Gaussian), leading to a loss of realism in the generated images. To address this, we introduce a sampling-based approach that balances reward maximization with adherence to the prior latent distribution. Specifically, we propose a novel method that simulates *Langevin dynamics*, where the drift term is determined by our orientation-grounding reward. We further show that its Euler–Maruyama discretization simplifies to a surprisingly simple formulation–an extension of standard gradient ascent with random noise injection, which can be implemented in a single line of code. To further enhance efficiency, we introduce a novel time-rescaling method that adjusts timesteps based on the current reward value, accelerating convergence.

Since no existing method has quantitatively evaluated 3D orientation grounding in text-to-image generation (except for user studies by [18]), we curate a benchmark based on the MS-COCO dataset [23], mixing and matching object classes and orientations to create images with single or multiple orientation-grounded objects. We demonstrate that ORIGEN significantly outperforms previous orientation-conditioned image generative models [18, 17] on both our benchmark and user studies. Since prior models cannot condition on multiple objects (whereas ORIGEN can, as shown in Fig. 1), comparisons are conducted under single-object conditioning. Additionally, we perform experiments to further validate the superior performance of our method over text-to-image generative models with orientation-specific prompts and other training-free guided sampling strategies.

Overall, our main contributions are:

- We present ORIGEN, the first method for 3D orientation grounding in text-to-image generation for multiple objects across diverse categories.
- We introduce a novel reward-guided sampling approach based on *Langevin dynamics* that provides a theoretical guarantee for convergence while simply adding a single line of code.
- We also propose a reward-adaptive time rescaling method that accelerates convergence.
- We show that ORIGEN achieves significantly better 3D orientation grounding than existing orientation-conditioned image generative models [18, 17], text-to-image generative models [24, 25, 21] with orientation-specific prompts, and training-free guided sampling strategies.

2 Related Work

Viewpoint or Orientation Control. Several works have focused on controlling the *global viewpoint* of the entire image. For example, Burgess *et al.* [26] propose a view-mapping network that predicts a word embedding to control the viewpoint in text-to-image generation. Kumari *et al.* [27] enable model customization to modify object properties via text prompts, with added viewpoint control. However, these methods cannot individually control the orientation of foreground objects. Other works have attempted to control *single-object orientation* in image generation. For instance, Liu *et*

al. [17] introduce an image diffusion model that controls the *relative* orientation of an object with respect to its reference image. Huang *et al.* [19] propose an orientation-conditioned image diffusion model for sampling multi-view object images for text-to-3D generation. The most recent work in this domain, Cheng *et al.* [18], aim to control object attributes, including azimuth, through continuous word embeddings. However, these methods rely on training-based approaches using single-object synthetic training images, limiting their generalizability across multiple objects and diverse categories. We additionally note that a few works address image generation conditioned on *depth* [28–30] or *3D bounding boxes* [31], but they do not allow direct control of object orientations — for example, 3D bounding boxes have front-back ambiguities.

Training-Free Guided Generation. A number of *training-free* methods have been proposed for guided generation tasks. DPS [32], MPGD [33], and Pi-GDM [34] update the noisy data point at each step using a given reward function. FreeDoM [35] takes this further by introducing *rewinding*, where intermediate data points are regenerated by reversing the generative denoising process. The core principle of this approach is leveraging the posterior mean from a noisy image at an intermediate step via Tweedie's formula [36]. The expected future reward can also be computed from the posterior mean, allowing gradient ascent to update the noisy image. However, a key limitation of these methods is that the posterior mean from a diffusion model is often too blurry to accurately predict future rewards. While distilled or fine-tuned diffusion [37–40] and flow models [41, 42] can mitigate this issue, their straightened trajectories lead to insufficiently small updates to the image during gradient ascent at intermediate timesteps.

Recent approaches [22, 43–45] attempt to address this limitation by updating the *initial* noise rather than the intermediate noise. Notably, ReNO [22] is an optimization method based on a *one-step* generative model to efficiently iterate the initial noise update through one-step generation and future reward computation. However, gradient ascent with respect to the initial noise often suffers from local optima and leads to deviations from real images, even with heuristic regularization [46]. To overcome this limitation, we propose a novel sampling-based approach rather than an optimization-based one, leveraging *Langevin dynamics* to effectively balance reward maximization and realism.

Training-Based Guided Generation. For controlling image generation, ControlNet [28] and IP-Adapter [47] are commonly used to utilize a pretrained generative model to control for various conditions, though they require training data. For 3D orientation grounding, no public real-world training data is available, and collecting such data would be particularly challenging, especially for multi-object grounding, due to the need for diversity. To address this, recent works have introduced RL-based reward fine-tuning approaches [48**?** –51]. However, these methods require substantial computational resources and, more importantly, cause significant deviation from the pretrained data distribution when trained with task-specific rewards, leading to degraded image quality and reduced diversity [52]. Hence, instead of fine-tuning, we propose a test-time reward-guided framework that leverages a discriminative foundational model for guidance.

3 ORIGEN

We present ORIGEN, a *zero-shot* method for 3D orientation grounding in text-to-image generation. To the best of our knowledge, this is the first 3D orientation grounding method for *multiple* objects across open-vocabulary categories.

3.1 Problem Definition and Overview

Our goal is to achieve 3D orientation grounding in text-to-image generation using a *one-step* generative flow model [53] $\mathcal{F}_\theta$, based on *test-time* guidance without fine-tuning the model. Let $\mathbf{I} = \mathcal{F}_\theta(\mathbf{x}, c)$ denote an image generated by $\mathcal{F}_\theta$ given an input text c and a latent $\mathbf{x}$ sampled from a prior distribution $q = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The input text c prompts the generation of an image containing a set of desired objects (e.g., "*A person* in a brown suit is directing *a dog*"). For each of the N objects that appear in c, we associate a set of object phrases $\mathcal{W} = \{w_i\}_{i=1}^N$ (e.g., {"*dog*", "*person*"}) and a corresponding set of 3D orientation grounding conditions $\Phi = \{\phi_i\}_{i=1}^N$. Following the convention [20], each object orientation ϕ_i is represented by three Euler angles $\phi_i = \{\phi_{i,j}\}_{j=1}^3$: azimuth $\phi_{i,1} \in [0, 360)$, polar angle $\phi_{i,2} \in [0, 180)$, and rotation angle $\phi_{i,3} \in [0, 360)$. The goal of 3D orientation grounding is to generate an image $\mathbf{I}$ in which the N objects appear with their corresponding orientations $\Phi = \{\phi_i\}_{i=1}^N$.

Why Test-Time Guidance? Supervised methods are not applicable to this task due to the lack of training datasets containing diverse real-world images with per-object orientation annotations. Previous supervised approaches [18, 17] have therefore been limited to single-object scenarios within a narrow set of categories. To address this, we propose a training-free approach that leverages recent foundational models to design a reward function: GroundingDINO [54] for object detection and OrientAnything [20] for orientation estimation. While this reward function could also be used in recent RL-based self-supervised methods for fine-tuning the image generative model, such techniques not only demand substantial computational resources but also lead to degraded image quality when trained with such a task-specific reward [52]. To this end, we propose a novel training-free approach that effectively avoids image quality degradation while maximizing orientation reward at test time.

In particular, given a reward function $\mathcal{R}$ defined based on GroundingDINO and OrientAnything (which we will discuss in detail in Sec. 3.2), our goal is to find a latent sample $\mathbf{x}$ that maximizes the reward of its corresponding image, expressed as $\mathcal{R}(\mathcal{F}_\theta(\mathbf{x}, c))$. For simplicity, we define the pullback of the reward function as $\hat{\mathcal{R}} = \mathcal{R} \circ \mathcal{F}_\theta$ and use this notation throughout.

Why Based on a One-Step Generative Model? Despite extensive research on test-time reward-guided generation using pretrained diffusion models [32, 33, 35], we find that the main challenge stems from the multi-step nature of these models. Reward-guided generation requires computing rewards for the expected final output at intermediate denoising steps and applying gradient ascent. However, in early stages, the expected output is too blurry to provide meaningful guidance, while in later stages—when the output becomes clearer—gradient updates have minimal effect. In contrast, a one-step model generates a clear image directly from a prior sample, enabling more effective guidance. Comparative results are presented in Section 4.

Our Technical Contribution. The gradient ascent approach for maximizing the future reward $\hat{\mathcal{R}}$ can also be applied to the one-step model, using the following update rule on the latent $\mathbf{x}_i$ at each iteration i:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \eta \nabla \hat{\mathcal{R}}(\mathbf{x}_i), \tag{1}$$

where η is a step size. However, this gradient ascent in the latent space pose several challenges: (1) the latent sample $\mathbf{x}$ may get stuck in local maxima [55, 56] before achieving the desired orientation alignment, (2) the mode-seeking nature of gradient ascent can reduce sample diversity [57], and (3) $\mathbf{x}$ may deviate from the prior latent distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, resulting in unrealistic images [48, 58]. Although the recent reward-guided noise optimization method (ReNO [22]) employs norm-based regularization [43, 59] to enforce the latent to be close to the prior distribution, it still suffers from local optima, leading to suboptimal orientation grounding results (see comparisons in Sec. 4 and further analysis in Appendix A).

Our key idea to address this is to reformulate the problem as a *sampling problem* rather than an *optimization problem*. Specifically, we aim to sample $\mathbf{x}$ from a target distribution q^* that maximizes the expected reward, while ensuring q remains close to the original latent distribution:

$$q^* = \arg\max_p \ \mathbb{E}_{\mathbf{x}\sim p}[\hat{\mathcal{R}}(\mathbf{x})] - \alpha D_{\mathrm{KL}}(p \,\|\, q), \tag{2}$$

where $\alpha \in \mathbb{R}$ is a constant that controls the regularization strength, and $D_{\mathrm{KL}}(\,\cdot \,\|\, \cdot\,)$ is the Kullback-Leibler divergence [60]. This objective is closely related to those used in fine-tuning-based approaches for reward maximization [61, 48, 62]. However, the key difference is that, while these methods define the target distribution q^* for the *output images*, we define it for the *latent samples*, setting q as the prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. This formulation leads to our key contribution: a novel, simple, and effective sampling method based on *Langevin dynamics*, discussed in Sec. 3.3.

Below, we first introduce our reward function designed for 3D orientation grounding (Sec. 3.2) and propose *reward-guided Langevin dynamics* to effectively sample from our target distribution q^* (Sec. 3.3). Lastly, we introduce *reward-adaptive time rescaling* to speed up sampling convergence by incorporating time scheduling (Sec. 3.4).

3.2 Multi-Object Orientation Grounding Reward

To define our reward function $\mathcal{R}$, we first detect the described objects in the generated image using GroundingDINO [54] with the provided object phrases $\mathcal{W}$, yielding cropped object regions $\mathrm{Crop}(\mathbf{I}, w_i)$ for each phrase w_i. We then use OrientAnything [20] to assess how well the orientations of the detected objects align with the specified grounding conditions Φ. OrientAnything represents

object orientations as categorical probability distributions over one-degree bins for each Euler angle. Let $\mathcal{D}_j(\cdot)$ denote the predicted distribution for the j-th angle. Following the convention from OrientAnything, we define the target distribution $\Pi(\phi_{i,j})$ as a discretized Gaussian centered at the reference angle $\phi_{i,j}$. The reward function $\mathcal{R}$ is then computed as the negative KL divergence between the predicted and target distributions, summed across all angles and all objects:

$$\mathcal{R}(\mathbf{I}, \mathcal{W}, \Phi) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j\in\mathcal{S}} D_{\mathrm{KL}}\Big(\mathcal{D}_j\big(\mathrm{Crop}(\mathbf{I}, w_i)\big) \,\Big\|\, \Pi(\phi_{i,j})\Big). \tag{3}$$

3.3 Reward-Guided Langevin Dynamics

We now introduce a method to effectively sample a latent $\mathbf{x}$ from our target distribution in Eq. 2. To address the limitations of vanilla gradient ascent for reward maximization (as discussed in Sec. 3.1), we propose enhancing the exploration of the sampling space of $\mathbf{x}$ by injecting *stochasticity*, which is known to help avoid local optima or saddle points [55, 56]. In particular, we show that simulating Langevin dynamics, in which the drift term is determined by our reward function (Sec. 3.2), can sample $\mathbf{x}$ that effectively aligns with the grounding orientation conditions.

Proposition 1. ***Reward-Guided Langevin Dynamics.*** *Let $q = \mathcal{N}(\mathbf{0}, \mathbf{I})$ denote the prior distribution, $\hat{\mathcal{R}}(\mathbf{x})$ be the pullback of a differentiable reward function, and $\mathbf{w}_t$ denote the standard Wiener process. As $t \to \infty$, the stationary distribution of the following Langevin dynamics*

$$d\mathbf{x}_t = \left(\frac{-\mathbf{x}_t + \frac{1}{\alpha}\nabla\hat{\mathcal{R}}(\mathbf{x}_t)}{2}\right)dt + d\mathbf{w}_t \tag{4}$$

coincides with the optimal distribution of Eq. 2.

Proof. See Appendix B. □

This demonstrates that simulating the Langevin stochastic differential equation (SDE) (Eq. 4) in the latent space ensures samples $\mathbf{x}$ are drawn from the target distribution q^*, balancing reward maximization with proximity to the prior distribution (Eq. 2). Using Euler–Maruyama discretization [63], we express its discrete-time approximation as:

$$\mathbf{x}_{i+1} \approx \mathbf{x}_i + \left(\frac{-\mathbf{x}_i + \frac{1}{\alpha}\nabla\hat{\mathcal{R}}(\mathbf{x}_i)}{2}\right)\delta t + \sqrt{\delta t}\epsilon_i, \tag{5}$$

where $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. By introducing the substitutions $\gamma \leftarrow \delta t$ and $\eta \leftarrow \frac{1}{2\alpha}$, the above expression (Eq. 5) simplifies to the following intuitive update rule:

$$\mathbf{x}_{i+1} = \sqrt{1-\gamma}\,\mathbf{x}_i + \gamma\eta\nabla\hat{\mathcal{R}}(\mathbf{x}_i) + \sqrt{\gamma}\epsilon_i. \tag{6}$$

Notably, this final update step (Eq. 6) is *surprisingly simple*. Compared to the update step in standard gradient ascent (Eq. 1), only the stochastic noise term and scaling factor are additionally introduced, which require *just a single additional line of code*. This update rule integrates exploration through noise while explicitly ensuring proximity to a prior distribution.

3.4 Reward-Adaptive Time Rescaling

While our reward-guided Langevin dynamics already enables effective sampling of $\mathbf{x}$ for 3D orientation grounding, we additionally introduce *reward-adaptive time rescaling* to further accelerate convergence via time rescaling. In Leroy *et al.* [64], a time-rescaled SDE is introduced with a *monitor function* $\mathcal{G}$ that adaptively controls the step size, along with a correction drift term $\frac{1}{2}\nabla\mathcal{G}(\hat{\mathcal{R}}(\mathbf{x}))dt$ to preserve the stationary distribution of the original SDE. By defining a new time variable τ and the corresponding process $\tilde{\mathbf{x}}_\tau = \mathbf{x}_{t(\tau)}$ with correction drift term $\frac{1}{2}\nabla\mathcal{G}(\hat{\mathcal{R}}(\tilde{\mathbf{x}}_\tau))d\tau$, our time-rescaled version of reward-guided Langevin SDE in Eq. 4 is expressed as:

$$d\tilde{\mathbf{x}}_\tau = \mathcal{G}(\hat{\mathcal{R}}(\tilde{\mathbf{x}}_\tau))\left(-\frac{1}{2}\tilde{\mathbf{x}}_\tau + \eta\nabla\hat{\mathcal{R}}(\tilde{\mathbf{x}}_\tau)\right)d\tau + \frac{1}{2}\nabla\mathcal{G}(\hat{\mathcal{R}}(\tilde{\mathbf{x}}_\tau))d\tau + \sqrt{\mathcal{G}(\hat{\mathcal{R}}(\tilde{\mathbf{x}}_\tau))}d\tilde{\mathbf{w}}_\tau, \tag{7}$$

where the original time increment is rescaled according to $dt = \mathcal{G}(\hat{\mathcal{R}}(\tilde{\mathbf{x}}_\tau))d\tau$ and $d\mathbf{w}_t =$

Algorithm 1 ORIGEN

Require: c (Prompt), $\mathcal{W}$ (Object phrase set), Φ (Grounding orientation set), $\mathcal{F}_\theta$ (One-step T2I model), $\mathcal{R}$ (Reward), $\mathcal{G}$ (Monitor), M (# Steps), η (Scale), γ (Step size)

Initialize $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \mathcal{R}_* = -\inf$
for $i = 0$ to $M - 1$ **do**
  $\hat{\mathcal{R}}_i = \mathcal{R}(\mathcal{F}_\theta(\mathbf{x}_i, c), \mathcal{W}, \Phi)$ ▷ *Reward Calc.*
  $\gamma_i = \mathcal{G}(\hat{\mathcal{R}}_i)\gamma$ ▷ *Timestep rescaling*
  $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_{i+1} = \sqrt{1-\gamma_i}\mathbf{x}_i + \gamma_i\eta\nabla\hat{\mathcal{R}}_i + \frac{1}{2}\gamma_i\nabla\log\mathcal{G}(\hat{\mathcal{R}}_i) + \sqrt{\gamma_i}\epsilon_i$ ▷ *Update step*
  if $\hat{\mathcal{R}}_i > \mathcal{R}_*$ **then**
    $\mathcal{R}_* = \hat{\mathcal{R}}_i, \mathbf{x}_* = \mathbf{x}_i$
  end if
end for
return $\mathcal{F}_\theta(\mathbf{x}_*, c)$

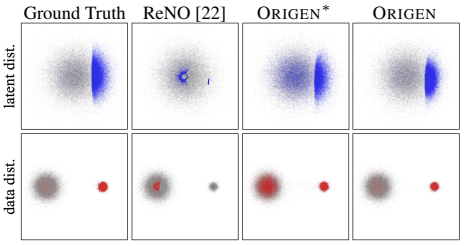


Figure 2: **Toy experiment results.** Top: latent space samples (blue); bottom: data space samples (red). Gray dots show the original distribution without reward guidance. From left to right: (1) ground truth target distribution from Eq. 2, (2) results of ReNO [22], (3) results of ours with uniform time scaling, and (4) results of ours with reward-adaptive time rescaling.

$\sqrt{\mathcal{G}(\hat{\mathcal{R}}(\tilde{\mathbf{x}}_\tau))}d\tilde{\mathbf{w}}_\tau$. A detailed derivation and convergence analysis of Eq. 7 are provided in Appendix C. Defining $\gamma(\tilde{\mathbf{x}}_\tau) = \mathcal{G}(\hat{\mathcal{R}}(\tilde{\mathbf{x}}_\tau))d\tau$, we obtain the following time-rescaled update rule via Euler-Maruyama discretization:

$$\mathbf{x}_{i+1} = \sqrt{1-\gamma(\mathbf{x}_i)}\mathbf{x}_i + \gamma(\mathbf{x}_i)\eta\,\nabla\hat{\mathcal{R}}(\mathbf{x}_i) + \frac{1}{2}\gamma(\mathbf{x}_i)\nabla\log\mathcal{G}(\hat{\mathcal{R}}(\mathbf{x}_i)) + \sqrt{\gamma(\mathbf{x}_i)}\,\epsilon_i. \tag{8}$$

Regarding the design of the monitor function, existing work [64] suggests setting the step size inversely proportional to the squared norm of the drift coefficients in Eq. 4. However, this approach requires computing the Hessian of the reward function for the correction term, which is computationally heavy. Alternatively, we propose following simple, reward-adaptive monitor function:

$$\mathcal{G}(\hat{\mathcal{R}}(\mathbf{x})) = s_{\text{min}} - \tanh(k\hat{\mathcal{R}}(\mathbf{x})) \cdot (s_{\text{max}} - s_{\text{min}}), \tag{9}$$

where the hyperparmeters s_{min}, s_{max}, and k are set to $\frac{1}{3}$, $\frac{4}{3}$, and 0.3 in our experiments, respectively. This function adaptively scales the step size by assigning smaller steps in high-reward and larger steps in low-reward, thereby improving convergence speed and accuracy. Please refer to Fig. 5 in Appendix that provides the visualization of our monitor function $\mathcal{G}(\hat{\mathcal{R}}(\mathbf{x}))$.

In Fig. 2, we present a toy experiment demonstrating the effectiveness of our Langevin dynamics and reward-adaptive time rescaling compared to gradient ascent with regularization (ReNO [22]). See Appendix D for details on the toy experiment setup. The top row shows the ground truth target latent distribution q^* (leftmost) alongside latent samples generated by different methods, and the bottom row displays the corresponding data distributions. While ReNO [22] fails to accurately capture the target distribution due to its mode-seeking behavior (2nd column, as discussed in Sec. 3.1), our method successfully aligns with it (3rd column), and time rescaling further accelerates convergence (4th column). Our sampling procedure is outlined in Alg. 1. Note that setting $\mathcal{G}(\hat{\mathcal{R}}(\mathbf{x})) = 1$ reverts the method to the uniformly time-rescaled form.

4 Experiments

4.1 Datasets

To the best of our knowledge, no existing benchmark has been proposed to evaluate 3D orientation grounding in text-to-image generation, aside from user studies conducted by Cheng *et al.* [18]. To address this, we introduce ORIBENCH based on the MS-COCO dataset [23], that consist of diverse text prompts, image, and bounding boxes.

ORIBENCH-Single. For a comparison with previous orientation-grounding methods [18, 17] that can condition on only a single object, we construct the ORIBENCH-Single dataset. From MS-COCO validation set [23], we filter out (1) object classes for which orientation cannot be clearly defined (e.g., objects with front-back symmetry) and (2) image captions lacking explicit object references. Since

Figure 3: **Qualitative comparisons on ORIBENCH-Single benchmark (Sec. 4.5).** Compared to the existing orientation-to-image models [18, 17], ORIGEN generates the most realistic images, which also best align with the grounding conditions in the leftmost column.

Figure 4: **Qualitative comparisons on ORIBENCH-Multi benchmark (Sec. 4.5).** Compared to the guided-generation methods [32, 33, 35, 22], ORIGEN generates the most realistic images, which also best align with the grounding conditions in the leftmost column.

the current orientation-to-image generation model [18] is only capable of controlling the front 180° range of azimuths, we further filter the samples to only include those within this range. This procedure yields 252 text-image pairs covering 25 distinct object classes. Using the provided bounding boxes, we cropped the foreground objects and fed them into OrientAnything [20], an orientation estimation model, to obtain pseudo-GT grounding orientations. Building upon this dataset, ORIBENCH-Single was constructed by mix-matching the image captions and grounding orientations, ultimately forming a dataset consisting of 25 object classes, each with 40 samples, totaling 1K samples (See Appendix E to check object classes we used).

ORIBENCH-Multi. For the comparison of a complex scenario with the grounding orientation of multiple objects, we construct ORIBENCH-Multi following an approach to that in ORIBENCH-Single. Since our base dataset [23] lacks samples composed solely of objects with a clear orientation, we mix-match object classes and grounding orientations from the 252 non-overlapping text-image pairs in ORIBENCH-Single, forming a dataset consisting of 371 samples, each containing a varying number of objects. We annotated prompts by concatenating individual object phrases (e.g., "a cat, and a dog.").

4.2 Evaluation Metrics

We compare ORIGEN's performance from two perspectives: (1) Orientation Alignment and (2) Text-to-Image Alignment. We measure orientation alignment using two metrics: 1) Acc.@22.5°,

Table 1: **Quantitative comparisons on 3D orientation grounded image generation.** Best and second-best results are highlighted in **bold** and underlined, respectively. ORIGEN* denotes ours without reward-adaptive time rescaling.

Id	Model	ORIBENCH-Single					ORIBENCH-Multi				
		Orientation Alignment		**Text-to-Image Alignment**			**Orientation Alignment**		**Text-to-Image Alignment**		
		Acc.@22.5° ↑	Abs. Err. ↓	CLIP ↑	VQA ↑	PickScore ↑	Acc.@22.5° ↑	Abs. Err. ↓	CLIP ↑	VQA ↑	PickScore ↑
(1) Fine-tuned Orientation-to-Image Model											
1	Zero-1-to-3 [17]	0.499	59.03	**0.272**	0.663	0.213	–	–	–	–	–
2	C3DW [18]	0.426	64.77	0.220	0.439	0.197	–	–	–	–	–
(2) Guided-Generation Methods with Multi-Step Model											
3	DPS [32]	0.664	28.16	0.246	0.662	0.221	0.487	38.28	0.285	0.742	0.231
4	MPGD [33]	0.689	27.62	0.246	0.661	0.222	0.532	35.03	0.283	0.739	**0.232**
5	FreeDoM [35]	0.741	20.90	0.259	0.728	**0.225**	0.591	31.77	0.279	0.783	0.227
(3) Guided-Generation Methods with One-Step Model											
6	ReNO [22]	0.796	20.56	0.247	0.663	0.212	0.478	45.95	0.277	0.756	0.223
7	**ORIGEN* (Ours)**	0.854	18.28	0.265	0.732	0.224	0.679	29.7	0.287	0.804	0.225
8	**ORIGEN (Ours)**	**0.871**	**17.41**	0.265	**0.735**	0.224	**0.692**	**28.4**	**0.287**	**0.807**	0.225

the angular accuracy within a tolerance of $\pm 22.5^\circ$ and 2) the absolute error on azimuth angles[1] between the predicted and grounding object orientations, following Wang *et al.* [20]. For evaluation, we use OrientAnything [20] to predict the 3D orientation from the generated images. Since its predictions may not be perfect, we also conduct a user study in Sec. 4.6 to validate the results. Along with grounding accuracy, we also evaluate text-to-image alignment using CLIP Score [65], VQA-Score [66], and PickScore [67].

4.3 Baselines

We compare ORIGEN with three types of baselines: (1) training-based orientation-to-image generation methods [18, 17] (2) training-free guided generation methods [35, 22] for the multi-step generation, and (3) training-free guided generation methods for the one-step generation. For (1), we include Continuous 3D Words (C3DW) [18] and Zero-1-to-3 [17], following the baselines used in the most recent work in this field [18]. For (2), we consider DPS [32], MPGD [33], and FreeDoM [35], which update the intermediate samples at each step of the multi-step sampling process based on the expected reward computed on the estimated clean image. Finally, for (3), we compare ORIGEN with ReNO [22], which optimizes the initial latent through vanilla gradient ascent using one-step models. Furthermore, to evaluate the effectiveness of our reward adaptive time rescaling, we perform additional comparison with our method variant without this component, denoted as ORIGEN*.

4.4 Implementation Details

We use FLUX-Schnell [21] as the one-step generative model for both ORIGEN and ReNO [22], while all multi-step guided generation baselines (DPS [32], MPGD [33], and FreeDoM [35]) are implemented using FLUX-Dev [21] as the multi-step generative model. Except for the training-based methods (C3DW [18] and Zero-1-to-3 [17]), we match the number of function evaluations (NFEs—defined as the number of iterations in our method and the denoising steps in multi-step generative models) to 50 for a fair comparison across all methods. All experiments were conducted on a single NVIDIA 48GB VRAM A6000 GPU. Further implementation details are provided in Appendix F.

4.5 Results

ORIBENCH-Single. In the left part of Tab. 1, we show our quantitative comparison results on the ORIBENCH-Single benchmark. ORIGEN *significantly* outperforms all the baselines in orientation alignment, showing comparable performance in text-to-image alignment. Our qualitative comparisons are also shown in Fig. 3, demonstrating that ORIGEN generates high-quality images that align with the orientation conditions and input text prompts. Note that C3DW [18] is trained on synthetic data (i.e., multi-view renderings of a 3D object) to learn orientation-to-image generation. Thus, it has limited generalizability to real-world images and the output images lack realism. Zero-1-to-3 [17] is also trained on single-object images but *without backgrounds*, requiring additional

[1]We perform comparisons only on azimuth angle, as existing methods [18, 17] do not support the control over polar and rotation angles. Note that our results for all azimuth, polar, and rotation angles are provided in Appendix G.1.

Table 2: **User Study Results.** 3D orientation-grounded text-to-image generation results of ORIGEN was preferred by 58.18% of the participants on Amazon Mechanical Turk [69], significantly outperforming the baselines [18, 17].

Method	Zero-1-to-3 [17]	C3DW [18]	**ORIGEN (Ours)**
User Preferences ↑ (%)	20.58	21.24	**58.18**

background image composition (also used in the evaluation of C3DW [18]) that may introduce unnatrual artifacts. The existing methods on guided generation methods also achieve suboptimal results compared to ORIGEN. In particular, all training-free multi-step guidance methods, including DPS [32], MPGD [33], and FreeDoM [35], perform poorly in orientation grounding. This is because object orientation control relies on modifying low-frequency image structures, which are primarily formed during early sampling stages, where multi-step diffusion models produce noisy and blurry outputs [68] (as discussed in Sec. 3.1). ReNO [22] also achieves suboptimal results compared to ours, as it performs latent optimization based on vanilla gradient ascent which is prone to local optima (as discussed in Sec. 3.1). Overall, our method achieves the best results both with and without time rescaling, demonstrating its ability to effectively maximize rewards while avoiding over-optimization.

ORIBENCH-Multi. We additionally show the quantitative comparison results on the ORIBENCH-Multi benchmark. Since no fine-tuned methods ((1) in Tab. 1) are capable of multi-object orientation grounding, we assess our method only with guided generation methods ((2), (3) in Tab. 1). Our quantitative results are presented in the right of Tab. 1, and qualitative examples are provided in Fig. 4. As shown, ORIGEN consistently outperforms all guided generation baselines under the same reward function. These results demonstrate that our object-agnostic reward design, combined with the proposed method, enables robust and generalizable orientation grounding even in complex multi-object scenarios.

Additional Results. We provide additional results for orientation grounding task (1) under more diverse orientations, and (2) for four primitive views (front, left, right, back) to enable comparisons with text-to-image models, where the orientation condition is included as an input text prompt. Please refer to Appendix G.

4.6 User Study

Our previous quantitative evaluations were performed by comparing the grounding orientations and orientations *estimated* from the generated images using OrientAnything [20]. While its orientation estimation performance is highly robust (as seen in all of our qualitative results), we additionally conduct a user study to further validate the effectiveness of our method based on human evaluation performed by 100 participants on Amazon Mechanical Turk [69]. Each participant was presented with the grounding orientation, the input prompt, and the images generated by (1) Zero-1-to-3 [17], (2) C3DW [18], and (3) ORIGEN. Since ORIGEN outperformed all guided-generation methods with same reward function design in quantitative comparison, they were excluded in the user study. Then, participants were asked to select the image that best matches both the grounding orientations and the input text prompt – directly following the user study settings in C3DW [18]. In Tab. 2, ORIGEN was preferred by 58.18% of the participants, clearly outperforming the baselines. For more details of this user study, refer to Appendix H.

4.7 Computation Time

In our experiments, we set the number of iterations in our method to match the denoising steps in multi-step generative models (e.g., 50), resulting in comparable computation time—approximately 52.7 seconds per image. A detailed comparison is provided in Appendix I. Notably, for cases achieving angular accuracy within a tolerance of $\pm 22.5^\circ$ (Tab. 3), the average number of iterations to reach this threshold was 12.8, corresponding to an average of 13.5 seconds. Our approach can also serve as a post-refinement method for existing models, potentially further reducing computation time.

5 Conclusion

We presented ORIGEN, the first 3D orientation grounding method for text-to-image generation across multiple objects and categories. To enable test-time guidance with a pretrained discriminator, we introduced a Langevin-based sampling approach with reward-adaptive time rescaling for faster convergence. Experiments showed that our method outperforms fine-tuning-based baselines and uniquely supports conditioning on multiple open-vocabulary objects.

Limitations and Societal Impact. Since our method relies on a pretrained discriminator, its performance is inherently bounded by the quality of that discriminator—though in our case, it still achieved state-of-the-art results by a significant margin. As a generative AI technique, our method may be misused to produce inappropriate or harmful content, underscoring the importance of responsible development and deployment.

Acknowledgement

This work was supported by the NRF of Korea (RS-2023-00209723); IITP grants (RS-2022-II220594, RS-2023-00227592, RS-2024-00399817, RS-2025-25441313, RS-2025-25443318, RS-2025-02653113); the National Program for Excellence in SW, supervised by the IITP; and the Technology Innovation Program (RS-2025-02317326), all funded by the Korean government (MSIT and MOTIE), as well as by the DRB-KAIST SketchTheFuture Research Center.

References

[1] Phillip Y Lee and Minhyuk Sung. ReGround: Improving textual and spatial grounding at no cost. In *ECCV*, 2024.

[2] Phillip Y. Lee, Taehoon Yoon, and Minhyuk Sung. GrounDiT: Grounding diffusion transformers via noisy patch transplantation. In *NeurIPS*, 2024.

[3] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: Open-set grounded text-to-image generation. In *CVPR*, 2023.

[4] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *CVPR*, 2023.

[5] Wan-Duo Kurt Ma, Avisek Lahiri, John P Lewis, Thomas Leung, and W Bastiaan Kleijn. Directed diffusion: Direct control of object placement through attention guidance. In *AAAI*, 2024.

[6] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. SpaText: Spatio-textual representation for controllable image generation. In *CVPR*, 2023.

[7] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[8] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing diffusion paths for controlled image generation. In *ICML*, 2023.

[9] Guillaume Couairon, Marlene Careil, Matthieu Cord, Stéphane Lathuiliere, and Jakob Verbeek. Zero-shot spatial layout conditioning for text-to-image diffusion models. In *ICCV*, 2023.

[10] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6D object pose estimation. In *CVPR*, 2019.

[11] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6D object pose prediction. In *CVPR*, 2018.

[12] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3D object coordinates. In *ECCV*, 2014.

[13] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6D object pose and size estimation. In *CVPR*, 2019.

[14] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. DenseFusion: 6D object pose estimation by iterative dense fusion. In *CVPR*, 2019.

[15] Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On evaluation of 6D object pose estimation. In *ECCV*, 2016.

[16] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D orientation learning for 6D object detection from rgb images. In *ECCV*, 2018.

[17] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *ICCV*, 2023.

[18] Ta-Ying Cheng, Matheus Gadelha, Thibault Groueix, Matthew Fisher, Radomir Mech, Andrew Markham, and Niki Trigoni. Learning continuous 3D words for text-to-image generation. In *CVPR*, 2024.

[19] Yuzhong Huang, Zhong Li, Zhang Chen, Zhiyuan Ren, Guosheng Lin, Fred Morstatter, and Yi Xu. OrientDream: Streamlining text-to-3D generation with explicit orientation control. *arXiv preprint arXiv:2406.10000*, 2024.

[20] Zehan Wang, Ziang Zhang, Tianyu Pang, Chao Du, Hengshuang Zhao, and Zhou Zhao. Orient Anything: Learning robust object orientation estimation from rendering 3D models. *arXiv preprint arXiv:2412.18605*, 2024.

[21] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[22] Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. ReNO: Enhancing one-step text-to-image models through reward-based noise optimization. In *NeurIPS*, 2024.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[24] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024.

[25] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *ECCV*, 2024.

[26] James Burgess, Kuan-Chieh Wang, and Serena Yeung-Levy. Viewpoint textual inversion: Discovering scene representations and 3D view control in 2d diffusion models. In *ECCV*, 2024.

[27] Nupur Kumari, Grace Su, Richard Zhang, Taesung Park, Eli Shechtman, and Jun-Yan Zhu. Customizing text-to-image diffusion with object viewpoint control. In *SIGGRAPH Asia*, 2024.

[28] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[29] Gyeongnyeon Kim, Wooseok Jang, Gyuseong Lee, Susung Hong, Junyoung Seo, and Seungryong Kim. DAG: Depth-aware guidance with denoising diffusion probabilistic models. *arXiv preprint arXiv:2212.08861*, 2022.

[30] Shariq Farooq Bhat, Niloy Mitra, and Peter Wonka. LooseControl: Lifting controlnet for generalized depth conditioning. In *SIGGRAPH*, 2024.

[31] Abdelrahman Eldesokey and Peter Wonka. Build-A-Scene: Interactive 3d layout control for diffusion-based image generation. In *ICLR*, 2025.

[32] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *ICLR*, 2023.

[33] Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *ICLR*, 2024.

[34] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *ICLR*, 2023.

[35] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-free energy-guided conditional diffusion model. In *ICCV*, 2023.

[36] Charles M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9 (6):1135–1151, 1981.

[37] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *ICML*, 2023.

[38] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency Trajectory Models: Learning probability flow ODE trajectory of diffusion. In *ICLR*, 2024.

[39] Shanchuan Lin, Anran Wang, and Xiao Yang. SDXL-Lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

[40] Weijian Luo, Zemin Huang, Zhengyang Geng, J. Zico Kolter, and Guo jun Qi. One-step diffusion distillation through score implicit matching. In *NeurIPS*, 2024.

[41] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[42] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. InstaFlow: One step is enough for high-quality diffusion-based text-to-image generation. In *ICLR*, 2024.

[43] Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-Flow: Differentiating through flows for controlled generation. In *ICML*, 2024.

[44] Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *ICCV*, 2023.

[45] Xiefan Guo, Jinlin Liu, Miaomiao Cui, Jiankai Li, Hongyu Yang, and Di Huang. InitNO: Boosting text-to-image diffusion models via initial noise optimization. In *CVPR*, 2024.

[46] Dvir Samuel, Rami Ben-Ari, Nir Darshan, Haggai Maron, and Gal Chechik. Norm-guided latent space exploration for text-to-image generation. In *NeurIPS*, 2023.

[47] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

[48] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *NeurIPS*, 2023.

[49] Fanyue Wei, Wei Zeng, Zhenyang Li, Dawei Yin, Lixin Duan, and Wen Li. Powerful and flexible: Personalized text-to-image generation via reinforcement learning. In *ECCV*, 2024.

[50] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023.

[51] Weijian Luo. Diff-Instruct++: Training one-step text-to-image generator model to align with human preferences. *arXiv preprint arXiv:2410.18881*, 2024.

[52] Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *ICLR*, 2025.

[53] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

[54] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *ECCV*, 2024.

[55] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *ICML*, 2011.

[56] Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, 2017.

[57] Rohit Jena, Ali Taghibakhshi, Sahil Jain, Gerald Shen, Nima Tajbakhsh, and Arash Vahdat. Elucidating optimal reward-diversity tradeoffs in text-to-image diffusion models. *arXiv preprint arXiv:2409.06493*, 2024.

[58] Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024.

[59] Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. In *AAAI*, 2024.

[60] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[61] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

[62] Hao Lang, Fei Huang, and Yongbin Li. Fine-tuning language models with reward learning on policy. *arXiv preprint arXiv:2403.19279*, 2024.

[63] Francesco Gianfelici. Numerical solutions of stochastic differential equations (kloeden, pk and platen, e.; 2008)[book reviews]. *IEEE Transactions on Neural Networks*, 19(11):1990–1991, 2008.

[64] Alix Leroy, Benedict Leimkuhler, Jonas Latz, and Desmond J Higham. Adaptive stepsize algorithms for langevin dynamics. *SIAM Journal on Scientific Computing*, 46(6):A3574–A3598, 2024.

[65] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[66] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *ECCV*, 2024.

[67] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *NeurIPS*, 2023.

[68] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *CVPR*, 2022.

[69] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, 2011.

[70] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598): 671–680, 1983. doi: 10.1126/science.220.4598.671.

[71] Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer-Verlag Berlin Heidelberg, 6 edition, 2003. ISBN 978-3-540-04758-2. doi: 10.1007/978-3-642-14394-6.

[72] Grigorios A. Pavliotis. *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*, volume 60 of *Texts in Applied Mathematics*. Springer, 2014. ISBN 978-1-4939-1322-0. doi: 10.1007/978-1-4939-1323-7.

[73] HuggingFace. OpenAI-CLIP, 2022.

[74] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[75] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment Anything. In *ICCV*, 2023.

[76] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, 2022.

[77] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024.

[78] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023.

[79] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. In *NeurIPS*, 2023.

[80] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *ECCV*, pages 439–457. Springer, 2024.

[81] Zehuan Huang, Yuan-Chen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. *arXiv preprint arXiv:2412.03632*, 2024.

[82] Eslam Mohamed Bakr, Pengzhan Sun, Xiaoqian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. In *ICCV*, 2023.

[83] Xiuquan Hou, Meiqin Liu, Senlin Zhang, Ping Wei, and Badong Chen. Salience detr: Enhancing detection transformer with hierarchical salience filtering refinement. In *CVPR*, 2024.

[84] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NeurIPS*, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification:

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Our two main datasets–ORIBENCH-Single and ORIBENCH-Multi (in Sec. 4.1–are carefully curated subsets of the MS-COCO validation set [23], specifically selected to support fine-grained evaluation of 3D orientation controllability in image generation. While these datasets are not currently released, they are constructed from publicly available datas using a reproducible filtering protocol, which we plan to release in future revisions. The code is also under preparation for public release.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to time and GPU constraints, we were unable to conduct a thorough statistical significance analysis prior to submission. However, we plan to include appropriate statistical evaluations (e.g., standard deviation, confidence intervals) and additional ablation studies in a future revision.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work builds on pretrained models and does not involve the release of new high-risk models or datasets. Therefore, specific safeguards were not applicable in this context.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: Although we introduce new benchmark datasets (ORIBENCH-Single and Multi), they are not publicly released at this stage. As such, they are not considered released assets for the purposes of this checklist item.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Our work does not use any large language models as part of the core methodology.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A Analysis on Norm-Based Regularization

In this section, we analyze the limitations of norm-based regularization in reward-guided noise optimization by highlighting its mode-seeking property, which can lead to overoptimization and convergence to local maxima. Consider the following reward maximization process with norm-based regularization [22, 46]:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \eta_1 \nabla \hat{\mathcal{R}}(\mathbf{x}_t) + \eta_2 \nabla \log y(\|\mathbf{x}_t\|), \tag{10}$$

where $\hat{\mathcal{R}} = \mathcal{R} \circ \mathcal{F}_\theta$ is the pullback of the reward function $\mathcal{R}$ and $y(\|\cdot\|)$ represents the probability density of a χ^d distribution.

We can view both $\hat{\mathcal{R}}(\cdot)$ and $y(\|\cdot\|)$ as components of a single reward, allowing us to rewrite the update as

$$\begin{aligned}
\mathbf{x}_{t+1} &= \mathbf{x}_t + \eta_1 \nabla \hat{\mathcal{R}}(\mathbf{x}_t) + \eta_2 \nabla \log y(\|\mathbf{x}_t\|) \\
&= \mathbf{x}_t + \eta_1 \nabla \hat{\mathcal{R}}(\mathbf{x}_t) + \eta_2 \nabla \Big[(d-1) \log \|\mathbf{x}_t\|^2 - \frac{1}{2} \|\mathbf{x}_t\|^2 \Big] \\
&= \mathbf{x}_t + \eta_1 \nabla \hat{\mathcal{R}}(\mathbf{x}_t) + \eta_2 \nabla K(\mathbf{x}_t) \\
&= \mathbf{x}_t + \nabla \Phi(\mathbf{x}_t)
\end{aligned} \tag{11}$$

where $\Phi(\mathbf{x}) = \eta_1 \hat{\mathcal{R}}(\mathbf{x}) + \eta_2 K(\mathbf{x})$. This optimization process is an Euler discretization (with $\delta t = 1$) of the ODE

$$\frac{d\mathbf{x}}{dt} = \nabla \Phi(\mathbf{x}), \tag{12}$$

which represents a deterministic gradient flow, whose stationary measure is a weighted sum of Dirac deltas located at the maximizers of $\Phi(\mathbf{x}) = \eta_1 \hat{\mathcal{R}}(\mathbf{x}) + \eta_2 K(\mathbf{x}) = 0$.

However, as illustrated in Fig. 2 in the main paper, this deterministic gradient ascent does not directly prevent the iterates from deviating substantially from the original latent distribution. Also, the process may collapse to local maxima–even ones where the reward is not sufficiently high [70]. Our empricial observations indicate that this approach is ineffective for our application, presumably because the reward function defined over the latent exhibits many local maxima, causing the deterministic ascent to converge to suboptimal solutions.

B Proofs

Proof of Proposition 1. Recall the standard result from overdamped Langevin dynamics: if $\mathbf{x}_t$ evolves according to the SDE:

$$d\mathbf{x}_t = \frac{1}{2} \nabla \log q^*(\mathbf{x}_t) dt + d\mathbf{w}_t, \tag{13}$$

then $q^*(\mathbf{x})$ is the unique stationary distribution. Using Eq. 2 in the main paper, we obtain

$$\frac{1}{2} \nabla \log q^*(\mathbf{x}) = -\frac{1}{2} \mathbf{x} + \frac{1}{2\alpha} \nabla \hat{\mathcal{R}}(\mathbf{x}_t). \tag{14}$$

Integrating this with respect to $\mathbf{x}$ gives

$$q^*(\mathbf{x}) \propto q(\mathbf{x}) \exp\left(\frac{\hat{\mathcal{R}}(\mathbf{x})}{\alpha} \right), \tag{15}$$

where $q(\mathbf{x})$ is a standard Gaussian distribution.

Finally, following existing approach [61], we easily arrive at

$$q^* = \arg\max_p \mathbb{E}_{\mathbf{x}\sim p}[\hat{\mathcal{R}}(\mathbf{x})] - \alpha D_{\mathrm{KL}}(p\|q), \tag{16}$$

which matches the expression presented in Eq. 2 in the main paper. □

C Analysis on Reward-Adaptive Time-Rescaled SDE

In this section, we analyze the effect of a position-dependent step size on reward-guided Langevin dynamics. We first illustrate why a naive time-rescaling approach fails to preserve the desired stationary distribution, and how to fix it with an additional correction term, eventually leading to Eq. 8 in the main paper. Our derivation follows the approach of Leroy *et al.* [64].

C.1 Non-Convergence of Direct Time-Rescaled SDE

Consider the following SDE:

$$d\mathbf{x}_t = b(\mathbf{x}_t)\, dt + \sigma(\mathbf{x}_t)\, d\mathbf{w}_t, \tag{17}$$

with drift $b(\mathbf{x}_t)$ and diffusion coefficient $\sigma(\mathbf{x}_t)$. Its probability density $\rho(\mathbf{x}, t)$ evolves according to the Fokker-Planck equation [71]:

$$\frac{\partial}{\partial t}\rho(\mathbf{x}, t) = -\nabla \cdot [b(\mathbf{x})\rho(\mathbf{x}, t)] + \frac{1}{2}\nabla^2\big[\sigma^2(\mathbf{x})\rho(\mathbf{x}, t)\big]. \tag{18}$$

Thus, the stationary distribution must lie in the *kernel* of the corresponding Fokker–Planck operator $\mathcal{L}^*$ [72]:

$$\mathcal{L}^*\rho(\mathbf{x}) = -\nabla \cdot \Big(b(\mathbf{x})\, \rho(\mathbf{x})\Big) + \frac{1}{2}\nabla^2\Big[\sigma^2(\mathbf{x})\, \rho(\mathbf{x})\Big]. \tag{19}$$

Now, consider the reward-guided Langevin SDE in Eq. 4 in the main paper:

$$d\mathbf{x}_t = \left(-\frac{1}{2}\mathbf{x}_t + \eta\, \nabla\hat{\mathcal{R}}(\mathbf{x}_t)\right) dt + d\mathbf{w}_t, \tag{20}$$

which has the stationary distribution $q^*(\mathbf{x})$ given by Eq. 2 in the main paper. By comparing with the general form, we identify:

$$b(\mathbf{x}) = -\frac{1}{2}\mathbf{x} + \eta\, \nabla\hat{\mathcal{R}}(\mathbf{x}), \quad \sigma(\mathbf{x}) = 1. \tag{21}$$

Hence,

$$\begin{aligned}\mathcal{L}^*q^*(\mathbf{x}) &= -\nabla \cdot \Big[\Big(-\frac{1}{2}\mathbf{x} + \eta\, \nabla\hat{\mathcal{R}}(\mathbf{x})\Big)\, q^*(\mathbf{x})\Big] + \frac{1}{2}\nabla^2\Big[q^*(\mathbf{x})\Big] \\ &= 0.\end{aligned} \tag{22}$$

Next, we introduce a monitor function $\mathcal{G}(\mathbf{x}) > 0$ and define a new time variable τ by

$$dt = \mathcal{G}\big(\mathbf{x}_{t(\tau)}\big)\, d\tau. \tag{23}$$

Defining the time-rescaled process $\tilde{\mathbf{x}}_\tau = \mathbf{x}_{t(\tau)}$, we rewrite the SDE as

$$d\tilde{\mathbf{x}}_\tau = \mathcal{G}(\tilde{\mathbf{x}}_\tau)\left(-\frac{1}{2}\tilde{\mathbf{x}}_\tau + \eta\, \nabla\hat{\mathcal{R}}(\tilde{\mathbf{x}}_\tau)\right) d\tau + \sqrt{\mathcal{G}(\tilde{\mathbf{x}}_\tau)}\, d\tilde{W}_\tau. \tag{24}$$

In this rescaled SDE, the new drift and diffusion coefficients are

$$a(\tilde{\mathbf{x}}) = \mathcal{G}(\tilde{\mathbf{x}})\left(-\frac{1}{2}\tilde{\mathbf{x}} + \eta\, \nabla\hat{\mathcal{R}}(\tilde{\mathbf{x}})\right), \quad \sigma(\tilde{\mathbf{x}}) = \sqrt{\mathcal{G}(\tilde{\mathbf{x}})}. \tag{25}$$

Table 3: **Convergence speed to the desired success criterion.** ORIGEN* denotes ours without reward-adaptive time rescaling.

Method	ORIGEN*	ORIGEN
Mean NFE ↓	14.2	**12.8**
Inference time (s) ↓	14.9	**13.5**

Applying the corresponding Fokker–Planck operator $\tilde{\mathcal{L}}^*$ to $q^*(\mathbf{x})$, we obtain

$$\begin{aligned}
\tilde{\mathcal{L}}^* q^*(\tilde{\mathbf{x}}) &= -\nabla \cdot \Big[\mathcal{G}(\tilde{\mathbf{x}})\Big(-\frac{1}{2}\tilde{\mathbf{x}} + \eta\,\nabla\hat{\mathcal{R}}(\tilde{\mathbf{x}})\Big) q^*(\tilde{\mathbf{x}})\Big] + \frac{1}{2}\nabla^2\Big[\mathcal{G}(\tilde{\mathbf{x}}) q^*(\tilde{\mathbf{x}})\Big] \\
&= -\nabla \cdot \Big[a(\tilde{\mathbf{x}})\, q^*(\tilde{\mathbf{x}})\Big] + \frac{1}{2}\nabla \cdot \Big[\nabla\mathcal{G}(\tilde{\mathbf{x}})\, q^*(\tilde{\mathbf{x}}) + \mathcal{G}(\tilde{\mathbf{x}})\,\nabla q^*(\tilde{\mathbf{x}})\Big] \\
&= -\nabla \cdot \Big[a(\tilde{\mathbf{x}})\, q^*(\tilde{\mathbf{x}})\Big] + \frac{1}{2}\nabla \cdot \Big[\nabla\mathcal{G}(\tilde{\mathbf{x}})\, q^*(\tilde{\mathbf{x}}) + 2a(\tilde{\mathbf{x}})\, q^*(\tilde{\mathbf{x}})\Big] \\
&= \frac{1}{2}\nabla \cdot \Big[\nabla\mathcal{G}(\tilde{\mathbf{x}})\, q^*(\tilde{\mathbf{x}})\Big] \\
&\neq 0,
\end{aligned} \tag{26}$$

where we used the fact that $\mathcal{G}(\mathbf{x})\,\nabla q^*(\mathbf{x}) = 2a(\mathbf{x})\, q^*(\mathbf{x})$, which follows from the identity $\nabla \log q^*(\mathbf{x}) = -\mathbf{x} + 2\eta\,\nabla\hat{\mathcal{R}}(\mathbf{x})$. Therefore, $q^*(\mathbf{x})$ is not annihilated by $\tilde{\mathcal{L}}^*$, implying that the time-rescaled SDE does not converge to the desired target distribution in Eq. 2 in the main paper.

C.2 Time-Rescaled SDE with Correction Drift Term

Convergence guarantee to original stationary distribution. Following previous work [64], we can add a correction drift term $\frac{1}{2}\nabla\mathcal{G}(\tilde{\mathbf{x}}_\tau)$ to Eq. 24 to ensure that the rescaled process converges to the same stationary distribution as the original SDE. The modified SDE becomes

$$d\tilde{\mathbf{x}}_\tau = \mathcal{G}(\tilde{\mathbf{x}}_\tau)\left(-\frac{1}{2}\tilde{\mathbf{x}}_\tau + \eta\,\nabla\hat{\mathcal{R}}(\tilde{\mathbf{x}}_\tau)\right) d\tau + \frac{1}{2}\nabla\mathcal{G}(\tilde{\mathbf{x}}_\tau) d\tau + \sqrt{\mathcal{G}(\tilde{\mathbf{x}}_\tau)}\, d\tilde{W}_\tau. \tag{27}$$

The corresponding Fokker-Planck operator $\tilde{\mathcal{L}}^*_{\text{corr}}$ for this corrected time-rescaled SDE now annihilates the original stationary distribution $q^*(\mathbf{x})$:

$$\begin{aligned}
\tilde{\mathcal{L}}^*_{\text{corr}} q^*(\tilde{\mathbf{x}}) &= -\nabla \cdot \Big[\mathcal{G}(\tilde{\mathbf{x}})\Big(-\frac{1}{2}\tilde{\mathbf{x}} + \eta\,\nabla\hat{\mathcal{R}}(\tilde{\mathbf{x}})\Big) q^*(\tilde{\mathbf{x}}) + \frac{1}{2}\nabla\mathcal{G}(\tilde{\mathbf{x}})\, q^*(\tilde{\mathbf{x}})\Big] + \frac{1}{2}\nabla^2\Big[\mathcal{G}(\tilde{\mathbf{x}}) q^*(\tilde{\mathbf{x}})\Big] \\
&= \frac{1}{2}\nabla \cdot \Big[\nabla\mathcal{G}(\tilde{\mathbf{x}})\, q^*(\tilde{\mathbf{x}})\Big] - \frac{1}{2}\nabla \cdot \Big[\nabla\mathcal{G}(\tilde{\mathbf{x}})\, q^*(\tilde{\mathbf{x}})\Big] \\
&= 0.
\end{aligned} \tag{28}$$

Consequently $q^*(\tilde{\mathbf{x}})$ remains in the kernel of the $\tilde{\mathcal{L}}^*_{\text{corr}}$, showing that this corrected time-rescaled SDE preserves the original invariant distribution.

Convergence speed of time rescaling approach. To verify the effectiveness of our time-rescaling approach, we analyzed the average number of iterations required to reach the desired success criterion during reward-guided Langevin dynamics on ORIBENCH-Single dataset. As shown in Tab. 3, ORIGEN* requires an average of *14.2* iterations to reach the desired reward level, whereas ORIGEN reduced this to *12.8* iterations, demonstrating improved convergence speed without image quality degradation. Notably, the computational overhead introduced by adaptive rescaling is negligible, as the term $\nabla \log \mathcal{G}(\hat{\mathcal{R}}_i)$ in Alg. 1 in the main paper can be efficiently computed via the chain rule, using the precomputed reward gradient $\nabla\hat{\mathcal{R}}_i$.

D Setup for the Toy Experiment

We train a *rectified flow model* [41] on a 2D domain, where the source distribution is $\mathcal{N}(0, \mathbf{I})$ and the target distribution is a mixture of two Gaussians, $\mathcal{N}(\mu_1, \sigma_1\mathbf{I})$ and $\mathcal{N}(\mu_2, \sigma_2\mathbf{I})$, with $\mu_1 =$

Class Number	Class Name
1	Airplane
2	Bear
3	Bench
4	Bicycle
5	Bird
6	Boat
7	Bus
8	Car
9	Cat
10	Chair
11	Cow
12	Dog
13	Elephant
14	Giraffe
15	Horse
16	Laptop
17	Motorcycle
18	Person
19	Sheep
20	Teddy bear
21	Toilet
22	Train
23	Truck
24	Tv
25	Zebra

Table 4: **Selected object classes in our dataset.**

$(4,0)^T, \mu_2 = (-4,0)^T, \sigma_1 = 0.3$, and $\sigma_2 = 0.9$. The velocity prediction network consists of four hidden MLP layers of width 128. We first train an initial model and distill it twice, resulting in a 3-rectified flow model. The reward function is defined as

$$\mathcal{R}(x,y) = \exp\left(-\frac{(x-4)^2+y^2}{2}\right) + 0.1 \cdot \exp\left(-\frac{(x+4)^2+y^2}{2}\right) - 1. \tag{29}$$

In Fig. 2 in the main paper, all methods use the same total number of sampling steps.

E Selected Classes

When constructing the ORIBENCH benchmark, we selected a total of 25 classes from the 80 object classes in MS-COCO validation set [23]. We excluded classes for which distinguishing the front and back is difficult or where defining a canonical orientation is ambiguous. The remaining 25 classes were chosen based on their clear orientation cues. The detailed list of selected classes is provided in Tab. 4.

F Implementation Details

Following the convention of OrientAnything [20], we set the standard deviation hyperparameters for the azimuth, polar, and rotation distributions to 20, 2, and 1, respectively, to transform discrete angles into the orientation probability distribution Π. For image quality evaluation, we utilized the official implementation of VQA-Score [66], and assessed CLIP Score [65], VQA-Score [66], and Pickscore [67] using OpenAI's CLIP-ViT-L-14-336 [73], LLaVA-v1.5-13b [74], and Pickscore-v1 [67] models, respectively.

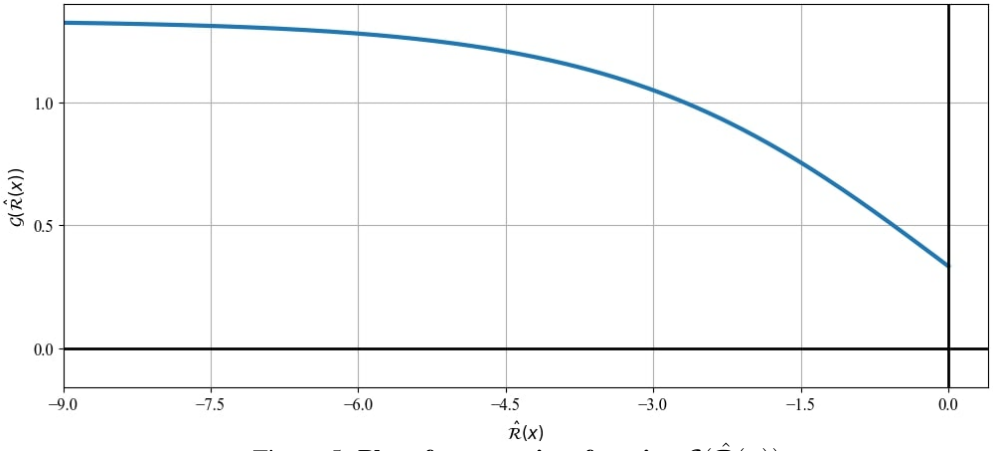


Figure 5: **Plot of our monitor function** $\mathcal{G}(\hat{\mathcal{R}}(\mathbf{x}))$.

About ORIGEN. We employed FLUX-Schnell [21] as our *one-step* T2I generative model and OrientAnything (ViT-L) [20] to measure the Orientation Grounding Reward, as detailed in Sec. 3.2. For all experiments, we set $\eta = 0.8$ in Alg. 1, and used $\gamma = 0.3$ for ORIBENCH-Single and $\gamma = 0.2$ for ORIBENCH-Multi, as this configuration provides a favorable balance between image quality and computational cost when evaluating with 50 NFEs. As described in Sec. 3.4, we additionally visualize our monitor function $\mathcal{G}(\hat{\mathcal{R}}(\mathbf{x}))$ from Eq. 9 in Fig. 5, which demonstrates how the function adaptively scales the step size–assigning smaller steps in high-reward regions and larger steps in low-reward regions–thereby improving convergence speed.

About Guided Generation Methods. For ReNO [22], we employed FLUX-Schnell [21] as the base one-step text-to-image generative model and OrientAnything (ViT-L) [20] as in ours. For multi-step guidance methods, we used FLUX-Dev [21] as multi-step text-to-image generative model. We set the gradient weight hyperparameter to 0.3 for ORIBENCH-Single and 0.2 for ORIBENCH-Multi for all methods. In the case of ReNO [22], we use norm-based regularization weight of 0.01. To ensure a fair comparison, we matched the number of function evaluations (NFEs) for all training-free one-step, multi-step guidance methods to 50.

About Zero-1-to-3 [17]. For the Zero-1-to-3 [17] baseline, we generated the base image using FLUX-Schnell (512×512) with 4 steps and encouraged the model to generate front-facing objects by adding the "facing front" prompt. The model was then conditioned on the target azimuth while keeping the polar angle fixed at 0° to generate novel foreground views, which were later composited with the background. The foreground was segmented using SAM [75], and missing background pixels were inpainted using LaMa [76] before composition.

About C3DW [18]. For C3DW [18], we utilized the orientation & illumination model checkpoint provided in the official implementation. This model is only capable of controlling azimuth angles for half-front views and was trained with scalar orientation values ranging from 0.0 to 0.5, where 0.0 corresponds to 90°, and 0.5 corresponds to -90°, with intermediate orientations obtained through linear interpolation. Using this mapping, we converted ground truth (GT) orientations into the corresponding input values for the model.

Handling Back-Facing Generation. We empirically observed that our base model struggled to generate back-facing images when the prompt did not explicitly contain view information. We hypothesized that this issue arises because high-reward samples lie within an extremely sparse region of the conditional probability space, making them inherently challenging to sample, even when employing our proposed approach. To address this issue, we explicitly added the phrase "facing back" in facing back cases (i.e., where $90 < \phi_i^{az} < 270$) and optimized the noise accordingly. With this simple adjustment, the model effectively generated both facing front and back images while maintaining high-quality outputs.

Table 5: **Quantitative comparisons on General Orientation Controllability.** ORIGEN maintains high accuracy even when evaluated on a more general set of samples. Best and second-best results are highlighted in **bold** and underlined, respectively.

Id	Model	Orientation Alignment						Text-to-Image Alignment		
		Azi, Acc. @22.5° ↑	Azi, Abs. Err. ↓	Pol, Acc. @5° ↑	Pol, Abs. Err. ↓	Rot, Acc. @5° ↑	Rot, Abs. Err. ↓	CLIP ↑	VQA ↑	Pick Score ↑
1	DPS [32]	0.432	43.89	0.511	13.07	0.969	1.66	0.259	0.704	**0.234**
2	MPGD [33]	0.398	48.09	0.485	12.25	0.967	1.79	0.258	0.707	**0.234**
3	FreeDoM [35]	0.470	38.09	0.554	12.71	**0.971**	**1.42**	0.261	0.709	0.229
4	ReNO [22]	0.586	41.41	0.502	14.37	0.958	2.69	0.253	0.676	0.216
5	**ORIGEN (Ours)**	**0.777**	**24.96**	**0.575**	**12.46**	0.969	1.52	**0.263**	**0.710**	0.219

Table 6: **Quantitative comparisons on 3D orientation grounded image generation for four primitive views.** Best and second-best results are highlighted in **bold** and underlined, respectively.

Id	Model	3-View (front, left, right)					4-View (front, left, right, back)				
		Orientation Alignment		Text-to-Image Alignment			Orientation Alignment		Text-to-Image Alignment		
		Acc.@22.5° ↑	Abs. Err. ↓	CLIP ↑	VQA ↑	PickScore ↑	Acc.@22.5° ↑	Abs. Err. ↓	CLIP ↑	VQA ↑	PickScore ↑
(1) One-step Text-to-Image Model											
1	SD-Turbo [24]	0.257	75.09	0.261	0.721	0.223	0.244	78.47	0.262	0.717	0.223
2	SDXL-Turbo [25]	0.189	78.44	0.265	0.722	0.227	0.196	81.88	0.262	0.717	0.227
3	FLUX-Schnell [21]	0.312	75.04	**0.268**	**0.739**	**0.230**	0.424	60.26	**0.267**	**0.739**	**0.229**
(2) Fine-tuned Orientation-to-Image Model											
4	Zero-1-to-3 [17]	0.366	59.03	0.266	0.646	0.210	0.321	75.10	0.264	0.642	0.209
5	C3DW [18]	0.504	64.77	0.187	0.334	0.188	–	–	–	–	–
(3) Guided-Generation Methods with One-Step Model											
6	**ORIGEN (Ours)**	**0.824**	**20.99**	0.262	0.721	0.220	**0.866**	**17.45**	0.262	0.720	0.220

G Additional Results on the Extended ORIBENCH-Single Benchmark

We further evaluate ORIGEN under two additional scenarios that better reflect real-world use cases, by extending the ORIBENCH-Single Benchmark. The first experiment investigates the question: (1) *"Can* ORIGEN *handle more general and complex orientations beyond simple angles?"*. The second experiment addresses: (2) *"Can orientation grounding be achieved simply by prompting a text-to-image generative model?"*. For the first scenario, we evaluate ORIGEN and other guided generation methods on an extended benchmark covering all three orientation components—azimuth, polar, and rotation without filtering the front range of azimuths. For the second scenario, we assess the capability of text-to-image models to perform orientation grounding for four primitive directions (front, left, right, back), where the orientation condition is provided via the input text prompt.

G.1 Orientation Grounding in more general and complex scenario.

We provide our extensive evaluation on general curated ORIBENCH-Single dataset. As discussed in Sec. 4.1, we filter out non-clear object classes and image captions, but we do not apply filtering on the front range and do not fix the polar and rotation angles. Upon this, we mix-match object classes and grounding orientations, forming a dataset consisting of 25 object classes, each with 40 samples, totaling 1K samples. We evaluated on same metrics as in Sec. 4.2 in the main paper, including polar and rotation accuracy, within a tolerance of ±5.0° as well as their absolute errors, following Wang *et al.* [20]. As shown in Tab. 5, ORIGEN generalizes well on diverse orientation conditions, outperforming all other training-free guidance methods. We also present qualitative comparisons for other training-free guidance methods in Fig. 6 . ORIGEN shows more consistent and accurate orientation alignment compared to other training-free approaches. This demonstrates the robustness of our approach, maintaining high orientation alignment performance even when evaluated on a more general and complex set of samples.

G.2 Text-to-Image Models.

As baselines, we consider several one-step T2I generative models: SD-Turbo [24], SDXL-Turbo [25], and FLUX-Schnell [21]. In particular, we appended a phrase that specifies the object orientation

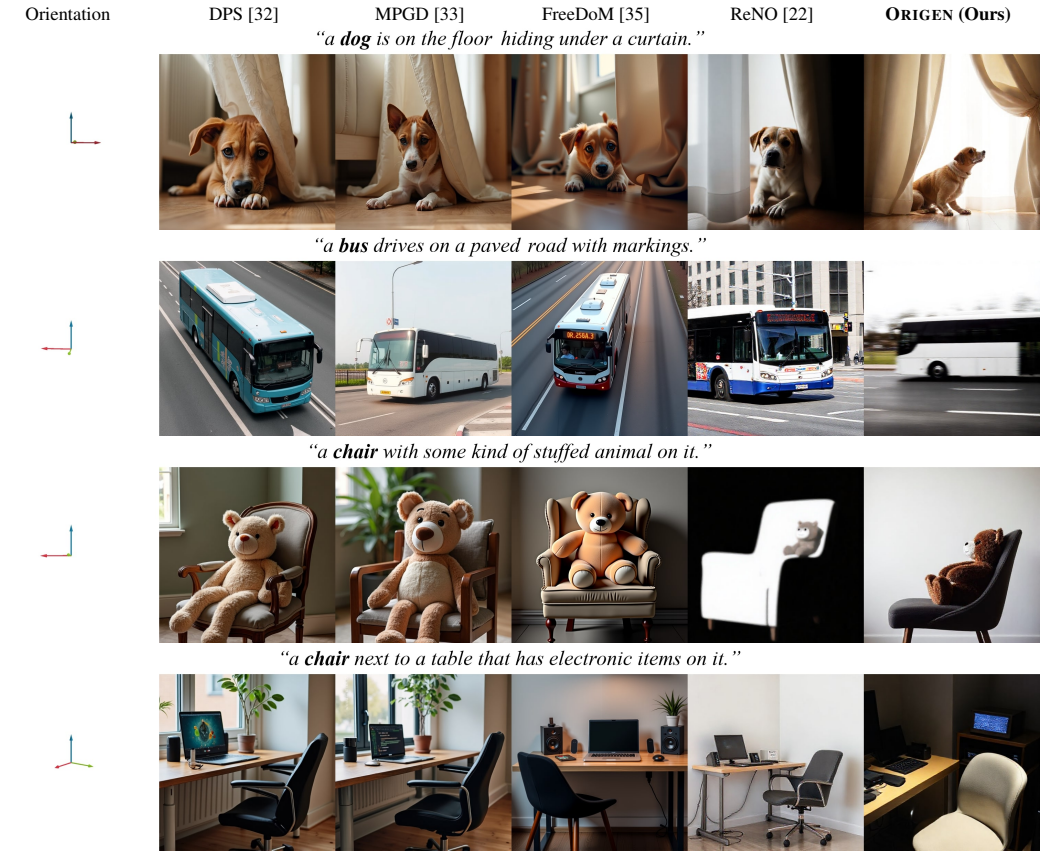
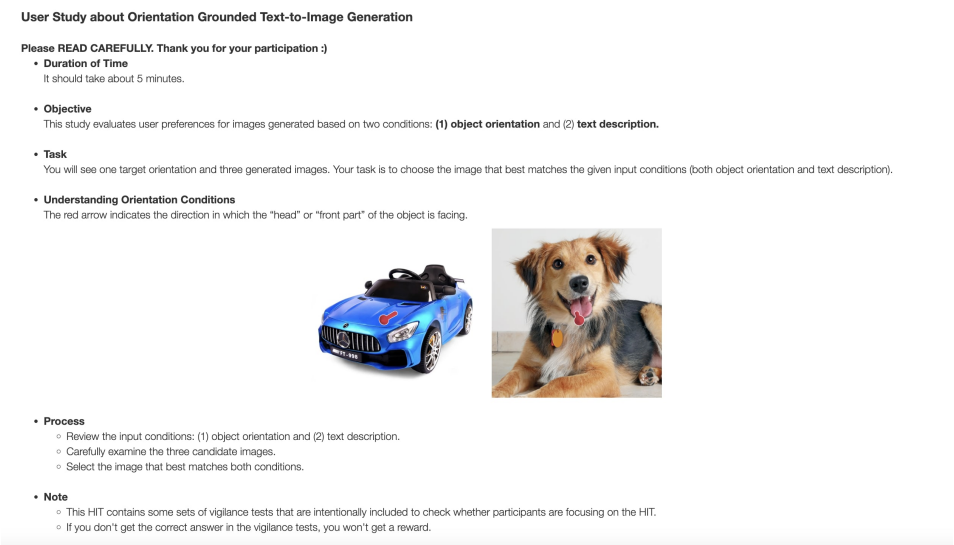
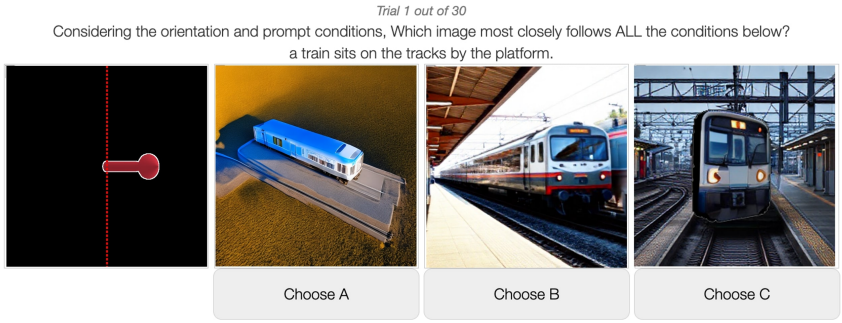


Figure 6: **Qualitative comparisons on generally extended ORIBENCH-Single benchmark.** Compared to other training-free approaches [32, 33, 35, 22], ORIGEN generates the best aligned images with the given orientation grounding conditions.

Figure 7: **Qualitative comparisons on extend ORIBENCH-Single benchmark for four primitive orientations.**



(a) Start page of the user study.



(b) Main test page of the user study.

Figure 8: **Screenshots of our user study.**

at the end of each caption in ORIBENCH-Single dataset. For more comprehensive comparisons, we also included the fine-tuned models (C3DW [18] and Zero-1-to-3 [17]) considered in Sec. 4.5 in this experiment as well. As shown in Tab. 6, ORIGEN significantly outperforms all baseline models in orientation alignment. Note that, although FLUX-Schnell [21] achieves the highest alignment among the vanilla T2I models, ORIGEN surpasses it by more than 2.5 times in the 3-view alignment setting (82.4% vs. 31.2%) and more than 2 times in the 4-view alignment setting (86.6% vs. 42.4%). This substantial margin highlights the inherent ambiguity and lack of precise control in vanilla T2I approaches, as orientation information embedded within textual descriptions is less explicit and reliable compared to direct orientation guidance. We further demonstrate the advantage of ORIGEN through qualitative comparisons in Fig. 7. Vanilla T2I models frequently fail to adhere strictly to the desired orientation, even with directional phrases in text prompts. In contrast, ORIGEN consistently generates images accurately aligned with the specified orientations.

H User Study Examples

In this section, we provide details of the user study. To assess user preferences, we conducted an evaluation comparing images generated by Zero-1-to-3 [17], C3DW [18], and ORIGEN using ORIBENCH-Single as the benchmark. The study was conducted on Amazon Mechanical Turk (AMT). For each object class, one sample was randomly selected, resulting in a total of 25 questions.

Table 7: **Computational cost comparison of orientation grounding methods on single NVIDIA A100 GPU.**

Method	NFEs	Time per Iter (Total)	VRAM	Img Size
SD-Turbo [24]	1	0.08s (0.08s)	4GB	512×512
SDXL-Turbo [25]	1	0.30s (0.30s)	15GB	1024×1024
FLUX-Schnell [21]	1	1.01s (1.01s)	32GB	512×512
C3DW [18]	20	6.09s (0.30s)	6GB	768×768
Zero-1-to-3 [17]	50	8.19s (0.16s)	46GB	256×256
DPS [32]	50	53.2s (1.06s)	43GB	512×512
MPGD [33]	50	39.3s (0.79s)	36GB	512×512
FreeDoM [35]	50	52.9s (1.06s)	43GB	512×512
ReNO [22]	50	54.5s (1.09s)	43GB	512×512
ORIGEN (Ours)	50	52.7s (1.05s)	43GB	512×512

Table 8: **Extended analysis on the reward-adaptive monitor function.** We ablate (i) the slope parameter k and (ii) the step-size bounds $(s_{\min}, s_{\max})$ to examine their effects on the sampling performance. For (i), $(s_{\min}, s_{\max})$ are fixed to their default values of $(\frac{1}{3}, \frac{4}{3})$, and for (ii), k is fixed to its default value of 0.3. Results indicate that overall performance is robust to variations in these hyperparameters. Best and second-best values are highlighted in **bold** and underlined, respectively.

(i) Ablation on slope parameter k

k	Azi. Acc. ↑	Azi. Err. ↓	CLIP ↑	VQA ↑	PickScore ↑
0.1	0.861	17.93	**0.267**	0.732	**0.226**
0.2	0.863	17.56	0.264	0.729	0.224
0.3	0.871	17.41	0.265	**0.735**	0.224
0.4	**0.882**	**17.17**	0.264	0.731	0.225
0.5	0.876	17.54	0.264	0.732	0.224

(ii) Ablation on step-size bounds $(s_{\min}, s_{\max})$

$s_{\min}$	$s_{\max}$	Azi. Acc. ↑	Azi. Err. ↓	CLIP ↑	VQA ↑	PickScore ↑
0.2	1.6	0.881	**16.72**	0.264	0.729	0.224
0.2	1.2	0.871	17.64	0.264	0.732	**0.225**
0.333	1.333	0.871	17.41	0.265	**0.735**	0.224
0.4	1.2	**0.887**	16.94	**0.266**	0.734	**0.225**
0.6	1.6	0.884	17.14	0.265	0.733	0.224

The orientations used to generate the images were intuitively visualized and presented alongside their corresponding prompts. As shown in Fig. 8, participants were asked to respond to the following question: *"Considering the orientation and prompt conditions, which image most closely follows ALL the conditions below?"*. Each user study session included 25 test samples along with 5 vigilance tests, which were randomly interspersed. The final results were derived from responses of valid participants who correctly answered at least three vigilance tests, leading to a total of 55 valid participants out of 100. No additional eligibility restrictions were imposed.

I Computational Costs

In Tab. 7, we compare the computational costs of all baselines and ORIGEN in terms of inference time and GPU memory consumption. Our base model (FLUX-Schnell [21]) and all guided generation methods were evaluated for generating 512×512 images, while other methods followed their default settings. Guided generation methods are generally slower due to the need for reward backpropagation. All methods were evaluated under a setup of 50 Number of Function Evaluations (NFEs). Further analysis demonstrating the fast convergence of our reward-guided Langevin Dynamics is provided in Appendix C.2, suggesting that the computational cost can be partially reduced by using fewer NFEs. All measurements were conducted on a single A100 GPU with 80GB of memory.

J Extended Analysis of the Reward-Adaptive Monitor Function

To further understand the effect of our reward-adaptive monitor function introduced in Sec. 3.4, we provide an extended analysis of its hyperparameters and their influence on model performance. As defined in Eq. 9, the monitor function adaptively modulates the sampling step size based on the reward, with three key hyperparameters: (i) k controlling the slope (how fast the step size decreases as the reward increases) and (ii) $(s_{\min}, s_{\max})$ defining the lower and upper bounds of the adaptive step size range.

We systematically vary these hyperparameters to evaluate the sensitivity of our sampling dynamics and its robustness to different configurations. Unless otherwise specified, all other hyperparameters are kept at their default values as reported in the main paper.

As shown in Tab. 8, our performance remains robust across all hyperparameter settings, with only negligible variations in the evaluation metrics. The default values were selected through a coarse grid search rather than fine-tuning, indicating that ORIGEN does not rely on fragile hyperparameter choices. Moreover, all variants with the reward-adaptive monitor further outperform the fixed-step baseline reported in Tab. 1 of the main paper (denoted as ORIGEN*), demonstrating its overall effectiveness.

K More Comprehensive Comparisons with Training-Based Methods

To complement the training-based results presented in Tab. 1 of the main paper, we provide more comprehensive quantitative comparisons with existing *training-based* approaches on the ORIBENCH-Single benchmark. Detailed descriptions of the baselines are provided below, and the quantitative results are summarized in the subsequent subsection.

K.1 Baselines.

Zero-1-to-3. [17] Please refer to 'About Zero-1-to-3' section in Appendix F.

C3DW. [18] Please refer to 'About C3DW' section in Appendix F.

DDPO. [77] We evaluated RL-based fine-tuning method for orientation alignment. Specifically, we adopt DDPO [77], which fine-tunes a diffusion model via policy gradient updates on a learned reward signal. We note that existing RL-based diffusion fine-tuning methods are designed for either unconditional or text-conditional generation [48, 49, 77]. Thus, to enable orientation conditioning while leveraging existing code of DDPO, we opt to appending orientation phrases to the input text prompts (e.g., "object is viewed from front-right, low-angle (azimuth 315°, polar 61°, rotation 90°)"). To fine-tune with our orientation grounding reward, we constructed a single-object text–orientation paired training dataset using 50k orientations from the OrientAnything [20] training dataset and captions from the Cap3D dataset [78, 79]. For other fine-tuning setups (e.g., hyperparameters), we primarily followed the default configuration of original DDPO to ensure fair comparisons.

SV3D. [80] SV3D is a multi-view diffusion model generates images conditioned on input image and target view angle. SV3D generates 21 views simultaneously, and we used the one generated image with target azimuth. The whole pipeline used in evaluation is same as Zero-1-to-3 [17], so please refer to 'About Zero-1-to-3' section in Appendix F.

MV-Adapter. [81] MV-Adapter is a multi-view diffusion model generates images conditioned on input image and target view angle. MV-Adapter generates 6 views simultaneously, and we used the one generated image with target azimuth. The whole pipeline used in evaluation is same as Zero-1-to-3 [17], so please refer to 'About Zero-1-to-3' section in Appendix F.

K.2 Results.

As shown in Table 9, ORIGEN achieves substantially higher orientation alignment accuracy than all training-based baselines while maintaining competitive text-to-image alignment performance. These results demonstrate that our inference-time reward-guided sampling consistently outperforms training-based methods that rely on explicit retraining or fine-tuning for orientation control.

L Reward-Guided Langevin Dynamics for Other Rewards

To assess the broader applicability of our reward-guided Langevin dynamics sampling, which is inherently reward-agnostic, we extended our experiments beyond 3D orientation grounding to additional tasks, including layout-to-image and depth-guided generation.

Table 9: **Quantitative comparisons on Training-based Methods.** ORIGEN maintains high accuracy even when compared with more recent training-based methods. Best and second-best results are highlighted in **bold** and underlined, respectively.

Id	Model	Orientation Alignment		Text-to-Image Alignment		
		Azi, Acc. @22.5° ↑	Azi, Abs. Err. ↓	CLIP ↑	VQA ↑	PickScore ↑
1	Zero-1-to-3 [17]	0.499	59.03	0.272	0.663	0.213
2	C3DW [18]	0.426	64.77	0.220	0.439	0.197
3	DDPO [77]	0.494	40.83	0.256	0.702	**0.233**
4	SV3D [80]	0.410	60.26	**0.274**	0.313	0.196
5	MV-Adapter [81]	0.486	50.19	0.204	0.313	0.196
6	**ORIGEN (Ours)**	**0.871**	**17.41**	0.265	**0.735**	0.224

Table 10: **Quantitative comparisons on layout-to-image and depth-guided generation.** Best and second-best results are highlighted in **bold** and underlined, respectively.

Id	Model	Layout-to-Image Generation					Depth-Guided Generation			
		Spatial Alignment		Text-to-Image Alignment			Depth Alignment	Text-to-Image Alignment		
		mIoU ↑	HRS ↑	CLIP ↑	VQA ↑	PickScore ↑	AbsRel ↓	CLIP ↑	VQA ↑	PickScore ↑
1	FreeDoM [35]	0.244	60	0.261	0.640	0.261	6.34	0.256	0.672	0.223
2	ReNO [22]	0.287	70	0.261	0.655	**0.229**	6.67	**0.265**	**0.705**	**0.233**
3	**ORIGEN (Ours)**	**0.344**	**72**	**0.284**	**0.659**	**0.229**	**4.62**	0.264	0.686	0.230

L.1 Experimental Setup

We used FreeDoM [35] and ReNO [22] as baselines to highlight the effectiveness of our sampling strategy.

Layout-to-Image Generation. For evaluation setup, we randomly sampled 100 examples from the HRS-Spatial dataset [82] and used GroundingDINO [54] as the reward model to guide object positioning based on bounding-box conditions. We evaluated the results using mIoU (measured by other detection model [83]) and HRS score.

Depth-Guided Generation. For evaluation setup, we randomly sampled 100 image-caption pairs from the ORIBENCH-Single dataset and generated pseudo-ground-truth depth maps using DepthAnything-V2 [84]. We then used DepthAnything-V2 as a reward model to guide image generation and evaluated the results using Absolute Relative Error (AbsRel).

L.2 Results

As shown in Fig. 9 and Tab. 10, ORIGEN consistently outperforms other guided-generation methods [35, 22] in condition alignment and image quality across both layout-to-image and depth-guided generation tasks. These results demonstrate that our method generalizes effectively to diverse reward alignment tasks.

M More Qualitative Comparisons on Single Object Orientation Grounding

In the following, Fig. 10 presents additional qualitative comparisons on the single object orientation grounding benchmarks.

N More Qualitative Comparisons on Multi Object Orientation Grounding

We report more qualitative results on multi object orientation grounding (ORIBENCH-Multi) in Fig. 11.

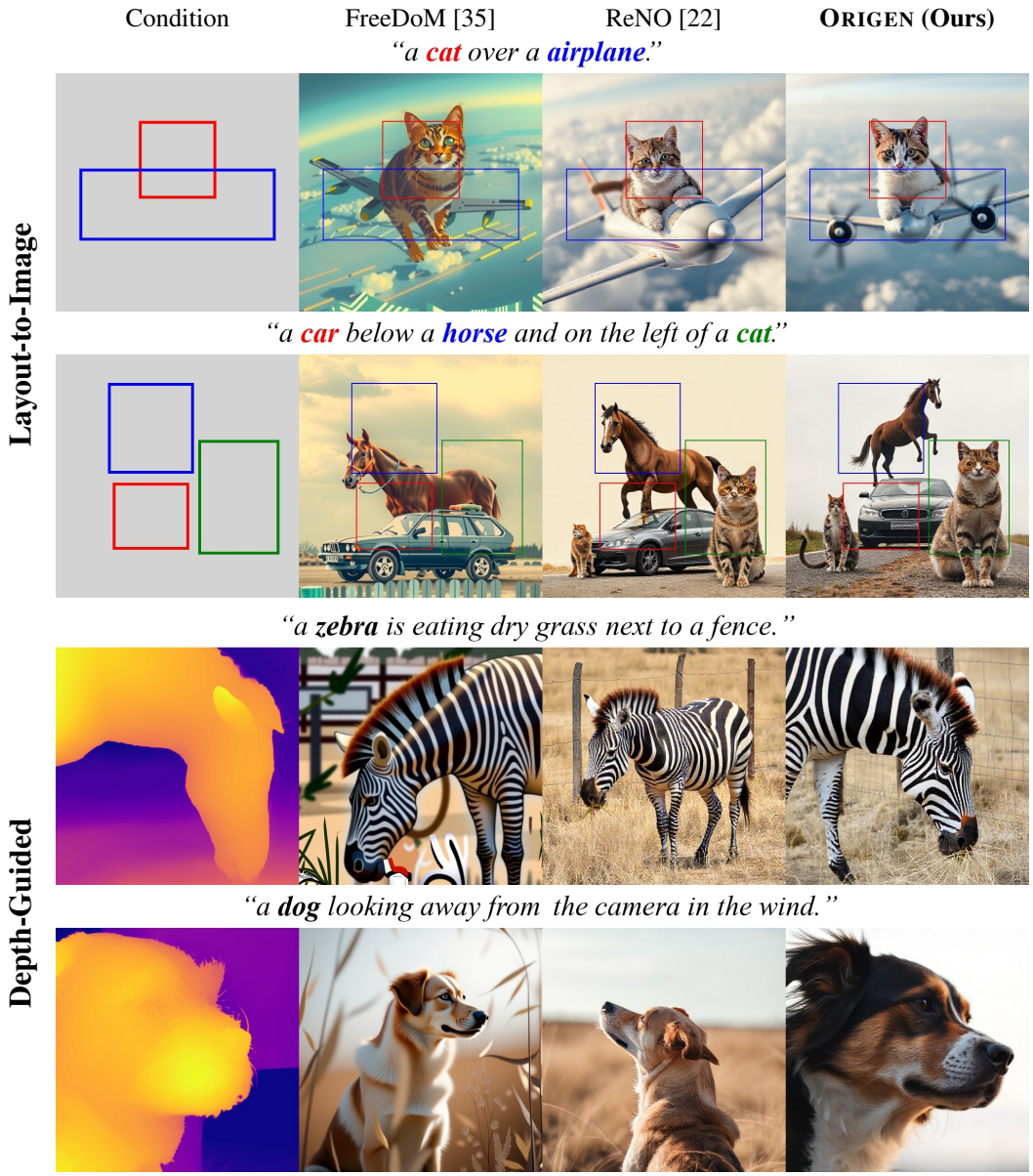


Figure 9: **Qualitative comparisons on layout-to-image (rows 1-2) and depth-guided generation (rows 3-4).** ORIGEN achieves better condition alignment and overall image quality compared to other guided-generation methods [35, 22].

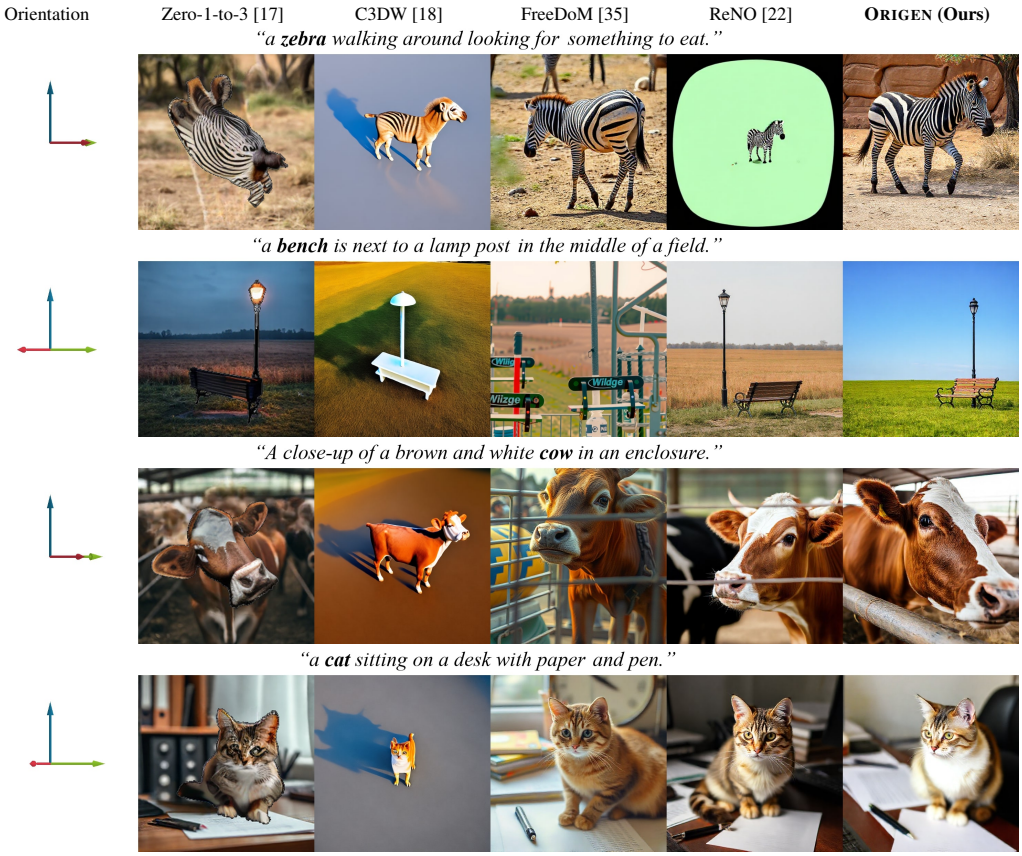


Figure 10: **Qualitative comparisons on ORIBENCH-Single benchmark (Sec. 4.5).** Compared to the existing orientation-to-image models [18, 17], ORIGEN generates the most realistic images, which also best align with the grounding conditions in the leftmost column.

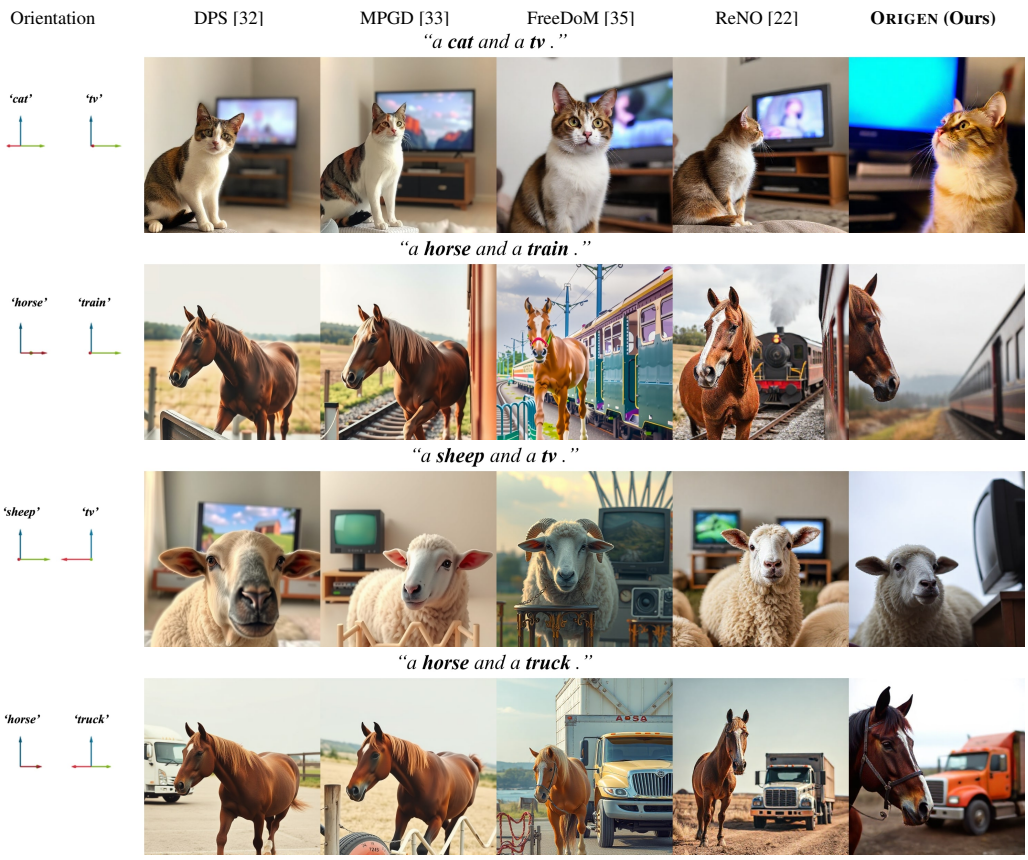


Figure 11: **Qualitative comparisons on ORIBENCH-Multi benchmark (Sec. 4.5).** Compared to the guided-generation methods [32, 33, 35, 22], ORIGEN generates the most realistic images, which also best align with the grounding conditions in the leftmost column.